

V.

Defendants.

**DEFENDANTS' JOINT RESPONSIVE  
CLAIM CONSTRUCTION BRIEF**

## TABLE OF CONTENTS

INTRODUCTION .....	1
FACTUAL BACKGROUND.....	2
A. Concurrent Licensing in the Prior Art .....	2
B. The ‘222 Patent .....	3
C. The Applicant’s Disclaimer and Clarification that the Daemon was “Independent of Separate from the Software Product[s]” .....	5
ARGUMENT .....	7
I. Disputed Claim Terms .....	7
A. “Usage tracking means” .....	7
B. Data Structure for a Policy Server Database, “Maintained Locally on the Given Computer” .....	10
C. “Policy Server Means” .....	13
1. “Daemon” is a Technical Term that Requires Explanation .....	15
2. Defendants’ Proposed Construction Properly Identifies the Correct Corresponding Structure .....	16
D. “Maintained and Operating Locally” .....	19
E. “As an Independent Process” .....	20
F. “Separate from the Software Product” .....	22
1. 112(6) Analysis .....	23
2. Ordinary Analysis .....	23
II. Agreed Constructions.....	25
CONCLUSION.....	25

## TABLE OF AUTHORITIES

### Cases

<i>Abbott Labs. v. Sandoz, Inc.</i> , 544 F.3d 1341, 1360 (Fed. Cir. 2008).....	16
<i>Bicon, Inc. v. Straumann Co.</i> , 441 F.3d 945, 950 (Fed. Cir. 2006).....	21, 22
<i>Biosig Instruments, Inc. v. Nautilus, Inc.</i> , 715 F.3d 891, 897, n.3 (Fed. Cir. 2013) .....	1
<i>Cardiac Pacemakers, Inc. v. St. Jude Med., Inc.</i> , 296 F.3d 1106, 1119 (Fed. Cir. 2002).....	11
<i>Default Proof Credit Card System, Inc. v. Home Depot U.S.A., Inc. (d/b/a The Home Depot)</i> , 412 F.3d 1291, 1298 (Fed. Cir. 2005).....	10, 20
<i>Dell USA L.P. v. Lucent Techs, Inc.</i> , 464 F.Supp.2d 620, 648 (E.D. Tex. 2006).....	19
<i>Elektta Instrument S.A. v. O.U.R. Scientific Intern., Inc.</i> , 214 F.3d 1302, 1307 (Fed. Cir. 2000).....	22
<i>Harris Corp. v. Ericsson Inc.</i> , 417 F.3d 1241, 1253 (Fed. Cir. 2005).....	16
<i>Home Diagnostics, Inc. v. LifeScan, Inc.</i> , 381 F.3d 1352, 1356 (Fed. Cir.2004).....	27
<i>Medical Instrumentation and Diagnostics Corp. v. Elektta AB</i> , 344 F.3d 1205, 1211 (Fed. Cir. 2003).....	9
<i>Multiform Desiccants, Inc. v. Medzam, Ltd.</i> , 133 F.3d 1473, 1477 (Fed. Cir. 1998).....	17
<i>Northrop Grumman Corp. v. Intel Corp.</i> , 325 F.3d 1346, 1352 (Fed. Cir. 2003).....	11, 20
<i>Omega Eng'g Inc. v. Raytek Corp.</i> , 334 F.3d 1314, 1323 (Fed. Cir. 2003).....	26
<i>Saffran v. Johnson &amp; Johnson</i> , 712 F.3d 549, 557, n.4 (Fed. Cir. 2013) .....	1

<i>Sta-Rite Indus., LLC v. ITT Corp.</i> , 682 F.Supp.2d 738, 744 (E.D. Tex. 2010).....	26
<i>Talbert Fuel Sys. Patents Co. v. Unocal Corp.</i> , 275 F.3d 1371, 1376 (Fed. Cir. 2002).....	14
<i>TV Interactive Data Corp. v. Sony Corp.</i> , Civil Action No. 10-cv-0475, 2012 WL 6020113, at *4 (N.D. Cal. Dec. 3, 2012).....	20
<i>WMS Gaming, Inc. v. Int’l Game Tech.</i> , 184 F.3d 1339, 1349 (Fed. Cir. 1999).....	16

#

## **Statutes**

35 U.S.C. ¶ 112(6) .....	1
--------------------------	---

#

## **Rules**

37 C.F.R. § 1.75(e).....	9
--------------------------	---

Pursuant to the Court’s Discovery Order (Dkt. 81) and P.R. 4-5(b), defendants Altair Engineering, Inc. (“Altair”), Altium, Inc. (“Altium”), and SlickEdit Inc. (“SlickEdit”) (collectively, “Defendants”) submit this brief in support of their proposed claim constructions and in response to Uniloc’s Opening Brief on Claim Construction (Dkt. 121) (“Pltf. Br.”).

## **INTRODUCTION**

The ‘222 Patent discloses a particular architecture and system for performing concurrent licensing. In particular, Claim 18 (the only claim asserted) discloses an improvement over prior art systems to address the complexity and burden of requiring each software product to use its own interface to contact the license server. Claim 18 describes a solution: a daemon, itself a piece of software, that sits on each node and operates as a “middle man” between the various software products on the node, on the one hand, and their respective network license servers, on the other. This middle man communicates with a local database and makes a permission-to-run determination at the node. The ‘222 Patent prescribes a particular set of algorithms, to which the patentee is bound by 35 U.S.C. ¶ 112(6), by which the middle man performs these functions.<sup>1</sup>

Plaintiff’s proposed claim construction ignores the plain meaning of the claim language and the basic principles of 35 U.S.C. ¶ 112(6), in an apparent effort to turn Claim 18 into a free-form “anything located anywhere doing anything” type of claim.

---

<sup>1</sup> Throughout, we cite the relevant pre-AIA provisions. See *Biosig Instruments, Inc. v. Nautilus, Inc.*, 715 F.3d 891, 897, n.3 (Fed. Cir. 2013) (“the pre-AIA version of § 112 applies because the [patent-in-suit] issued prior to that date”); *Saffran v. Johnson & Johnson*, 712 F.3d 549, 557, n.4 (Fed. Cir. 2013) (referring to the pre-AIA provisions of 35 U.S.C. § 112(f) when interpreting a patent issued before the AIA’s date of effectiveness).

## **FACTUAL BACKGROUND**

### **A. Concurrent Licensing in the Prior Art**

Software products are often licensed rather than sold.<sup>2</sup> That is, the product may be physically installed/present on the user's computer, but will only run when given permission in the form of a license, password, or other form of authorization.<sup>3</sup>

Where the computer is part of a network, use of a software product may be governed by a "concurrent licensing" system.<sup>4</sup> Concurrent licensing systems determine and limit the number of computers ("nodes") that may access a software product at a given time.<sup>5</sup> A license server on the network keeps track of how many computers are running local copies of the software, and distributes licenses as individual users request permission to use the software product.<sup>6</sup> If at any given time, the permitted number of licenses would be exceeded by a computer requesting a license, that computer is denied permission to use the software product. When a user stops running a piece of software on their computer, that license is returned to the license server, which can then be issued to another computer.<sup>7</sup>

---

<sup>2</sup> '222 Patent at 1:21-24.

<sup>3</sup> '222 Patent at 1:36-42.

<sup>4</sup> '222 Patent at 2:36-41.

<sup>5</sup> '222 Patent at 2:36-41.

<sup>6</sup> '222 Patent at 2:36-41.

<sup>7</sup> For example, a network owner might buy a concurrent license permitting up to ten users to run a Computer Aided Design (CAD) software product simultaneously. As long as there are fewer than ten users utilizing the CAD software at the same time, the license server can grant a license over the network to another computer requesting permission to use the CAD software. Once ten users are running the CAD, a request from an eleventh user would be denied (until one of the original ten stopped using the software).

## B. The '222 Patent

The '222 Patent discloses a particular architecture for performing concurrent licensing. The architecture uses a standard arrangement of networked nodes and a license server, *see id.* at 2:40-43 (describing prior art). The purported novelty is the placement of two additional components on the user's computer. The first component, a local policy server database (or "rules-based policy file"), specifies all of the conditions under which the user's computer may use the software products on the node."<sup>8</sup> The second component, a local daemon, makes a permission-to-run availability determination by reference to the local policy server database.<sup>9</sup> Figure 1 illustrates these two allegedly new components as elements 14 and 16:

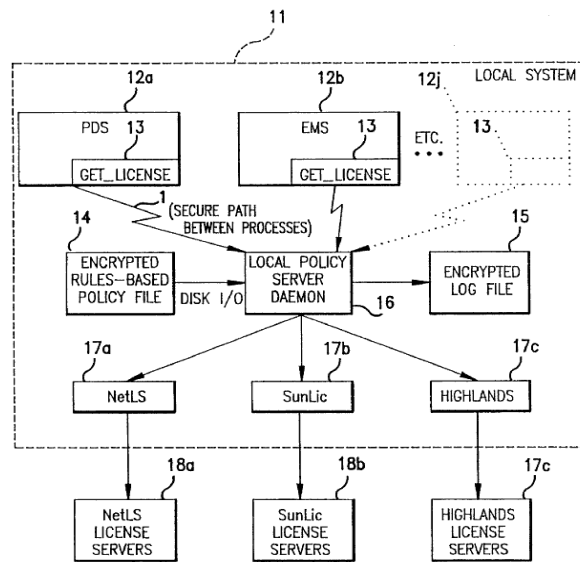


FIG.1

The first novel component, the policy server database, contains the rules under which a

<sup>8</sup> '222 Patent at 4:6-40.

<sup>9</sup> '222 Patent at 5: 39-44.

license may be granted to use the software product.<sup>10</sup> For example, the policy server database contains rules pertaining to the cost to use a software product,<sup>11</sup> prerequisite hardware requirements for the computer requesting the license,<sup>12</sup> and how to handle unexpected network and/or license server failures that prevent obtaining a license from the license server.<sup>13</sup>

The local daemon, itself a piece of software, resides on each computer node and operates as the middle man between the software products (Fig. 1 at 12a, 12b, etc.), on the one hand, and the respective network license servers 18a, 18b and 17c, on the other. Claim 18, which refers to the daemon as a “policy server means,” states that this “means” must be separate from the software products it serves:

policy server means, maintained and operating totally, as an independent process, *separate from the software product* on the given computer, and in association with the policy server database, for (i) communicating with the license server, (ii) interfacing with both (aa) the software product and (bb) the policy server database, and (iii) making a permission-to-run availability determination, with respect to local usage of the software product . . . .

‘222 Claim 18(b) (emphasis added). The asserted claim, in short, requires that each node (or computer) have a daemon that sits and operates *outside* the individual software products. The daemon is not a part of those products. It stands apart from them. It communicates with them. *See also, e.g.*, ‘222 Patent at Abstract (each node has a daemon). It acts as a middle man, sparing each software product the trouble of requesting a license from the license server itself. *See id.* at 3:3-7 (describing how, in the prior art, each software product would have to have a

---

<sup>10</sup> ‘222 Patent at 3:39-44.

<sup>11</sup> ‘222 Patent at 8:19-25.

<sup>12</sup> ‘222 Patent at 8:23-25.

<sup>13</sup> ‘222 Patent at 8:65 – 9:1.

“substantial software portion” dedicated to licensing).

**C. The Applicant’s Disclaimer and Clarification that the Daemon was  
“Independent of Separate from the Software Product[s]”**

All of the foregoing components were known in the prior art. U.S. Patent No. 5,138,712 (“Corbin”) disclosed the use of a “licensing library” software module that “communicated with the license server [via] a remote procedure call mechanism,” *see* Corbin at 6:40-43, and also interfaced with both usage rules and a software application. *See id.* at 6:40-42. The Corbin module also made a permission-to-run determination. Corbin at 8:63-64. Indeed, during prosecution, the applicant had to amend his claims to overcome Corbin. Claim 18(b) originally read:

policy server means, maintained and operating locally, on a given computer, and in association with the policy server database, for (i) communicating with the license server; (ii) interfacing with both (aa) the software product and (bb) the policy server database, and (iii) making a permission-to-run availability determination, with respect to local usage of the software product . . . .

JA00035 (file history, application dated December 14, 1992).

On April 5, 1995 the PTO rejected all of the claims in the application.<sup>14</sup> JA00271 (file history, April 5, 1995 office action). The Examiner stated that six of the claims were anticipated by Corbin. JA00273 (file history, April 5, 1995 office action). Of specific concern to the PTO was the “policy server means” claimed throughout the ‘222 Application. JA00276 (stating that Corbin “specifically detailed the claimed policy server means”) (internal citations omitted).<sup>15</sup>

---

<sup>14</sup> The PTO had rejected the claims before, on other grounds.

<sup>15</sup> Claim 18 was not expressly rejected by the PTO in its April 5, 1995 office action on anticipation grounds. JA00273-75 (file history, April 5, 1995 office action). Claim 18 did, however, claim the same “policy sever means” the PTO found to be anticipated by the ‘712 Patent. *See* JA00264 (file history, January 30, 1995 response to office action). It appears that it was not until a telephone interview with the examiner just prior to the applicants’ July 13, 1995

In response, the applicants amended several claims, including claim 18. JA00289-94 (file history, July 13, 1995 response to office action). The applicants added two phrases to the claim: “as an independent process” and “separate from the software product.” JA00294. After this amendment, sub-part (b) read:

policy server means, maintained and operating locally, as an independent process, separate from the software product, on a given computer, and in association with the policy server database, for (i) communicating with the license server; (ii) interfacing with both (aa) the software product and (bb) the policy server database, and (iii) making a permission-to-run availability determination, with respect to local usage of the software product...

*Id.* In their remarks, the applicants disclaimed any embodiment of their invention where the “policy server means,” or daemon was located within or ran as a part of a software product.

They explained that the amendment “make[s] clear that the policy server means, which operates locally as an independent process, is separate from the software product.” JA00294-95.

Specifically, the applicants argued that their invention was different from (and superior to) Corbin because it permitted a single daemon to service multiple software products:

Nothing in Corbin discloses or suggests making Corbin’s license access module run as a process independent of the other applications; Corbin specifically teaches that the module is application specific. A computer running five applications would have five of Corbin’s license access modules, one in each application. In contrast, in the case of the present invention, a single policy server means would function independently of all five applications.

Unlike the software-product-specific modules in Corbin, this new daemon would exist on its own and be available to any and all of the (e.g. five) software products on the node.

---

response that concerns regarding claim 18 and anticipation by the ‘712 Patent came to light. *See* JA00294 (file history, July 13, 1995 response to office action)(expressing appreciation for “the

The PTO issued a Notice of Allowance on June 5, 1996 based upon the applicant's July 13, 1995 amendments. JA00304 (file history, notice of allowance). With regard to overcoming the prior art, the PTO stated that "[t]he reasons for allowance over the prior art as stated by the applicant are persuasive when the combination of the elements in the claims are considered together." *Id.* at 305. The '222 Patent issued on November 26, 1999. '222 Patent at 1.

## **ARGUMENT**

### **I. DISPUTED CLAIM TERMS**

#### **A. "Usage tracking means"**

<b>Term/Clause</b>	<b>Plaintiff's Construction</b>	<b>Defendants' Construction</b>
"(ii) usage tracking means, associated with one of the computers acting as a license server, for (i) causing storage of the number of licenses available for running the software product on nodes of the network, (ii) identifying the current set of nodes with respect to which a license has been granted to run the software product at a given time, and (iii) determining whether at any given time any licenses remain to be granted for permitting an additional node to run the software product,"	<b><u>AGREED Function:</u></b> (1) causing storage of the number of licenses available for running the software product on nodes of the network; (2) identifying the current set of nodes with respect to which a license has been granted to run the software product at a given time; and (3) determining whether at any given time any licenses remain to be granted for permitting an additional node to run the software product.	
	<b><u>Structure:</u></b> computer memory in communication with the license server that contains a concurrent licensing program such as NetLS, SunLic, and Flexible License Manager, as disclosed at 2:36-3:11; 3:37-46 and 5:44-54, and equivalents.	<b><u>Structure:</u></b> a license server that contains a concurrent licensing program such as NetLS, SunLic, and Flexible License Manager, as disclosed at 2:36-3:11; 3:37-46 and 5:44-54, and equivalents.

The parties dispute whether "computer memory" (Plaintiff's proposal) or a "license server" of the type disclosed in the specification (Defendants' proposal) performs the agreed tripartite function. The issue boils down to what is disclosed and clearly linked to the claimed functions and attention of the examiner in a telephone interview conducted on July 12, 1995, during which the claims in relation to the Corbin reference were discussed").

whether the proposed structure is capable of performing them. *See Medical Instrumentation and Diagnostics Corp. v. Elekta AB*, 344 F.3d 1205, 1211 (Fed. Cir. 2003) (“structure disclosed in the specification is corresponding structure only if the specification or prosecution history clearly links or associates that structure to the function recited in the claim”). Plaintiff’s proposal identifies a structure (computer memory) not disclosed in the specification, not linked to the function, and, in fact, incapable of performing it.

The only disclosed structure clearly linked to the claimed function is a license server having a concurrent licensing program “such as NetLS”:

The improved system is of the type having an arrangement, such as NetLS, for tracking software product usage . . . This arrangement permits the license server (i) to identify the current set of nodes . . . (ii) to handle license data . . . and (iii) to determine whether at any given time the conditions would still be satisfied if a given node is then added . . .

‘222 Patent at 3:37-46; *see also* ‘222 Patent at 44-54 (“three separate license servers are shown: one (item 18a) running NetLS; another (item 18b) SunLic; and another (item 18c) Highlands”). These prior art licensing systems form the foundation for the ‘222 Patent’s improvement and perform the claimed function. *See, e.g.*, ‘222 Patent at 3:34-40 (“the present invention provides an improved system for administration”). Indeed, claim 1, which is drafted in the form suggested by 37 C.F.R. § 1.75(e) for improvement claims, recites as a known element in the preamble a license server for performing the same tripartite function as claim 18.<sup>16</sup> The ‘222 Patent specification clearly links the

---

<sup>16</sup> 37 C.F.R. § 1.75(e) provides:

(e) Where the nature of the case admits, as in the case of an improvement, any independent claim should contain in the following order:

(1) A preamble comprising a general description of all the elements or steps of the claimed combination which are conventional or known,

license server running one of these prior art licensing systems to the claimed function.

Despite describing the patent as having the license server perform these functions,<sup>17</sup> Plaintiff contends some generic “computer memory in communication with the license server” should be “fleshed out” and read into the claim. Pltf. Br. at 4 (acknowledging computer memory is at best “inherently disclosed”). Plaintiff’s proposal improperly rewrites the claim to identify and highlight structure that cannot even perform the recited functions. *Default Proof Credit Card System, Inc. v. Home Depot U.S.A., Inc. (d/b/a The Home Depot)*, 412 F.3d 1291, 1298 (Fed. Cir. 2005) (“While corresponding structure need not include all things necessary to enable the claimed invention to work, it must include all structure that actually performs the recited function.”) (citing *Cardiac Pacemakers, Inc. v. St. Jude Med., Inc.*, 296 F.3d 1106, 1119 (Fed. Cir.2002)).

Claim 18 recites “usage tracking means, associated with one of the computers acting as a license server.” Contrary to Plaintiff’s proposal, usage tracking means is not “computer memory” in communication with the license server”; it is “a computer program, acting as ‘librarian’ and running on a . . . license server.” ‘222 Patent at 2:41-50 (describing the server program performing the claimed function). Plaintiff cites no intrinsic support linking “computer memory” to the recited

---

(2) A phrase such as “wherein the improvement comprises,” and

(3) Those elements, steps and/or relationships which constitute that portion of the claimed combination which the applicant considers as the new or improved portion.

Claim 18 similarly set forth the “usage tracking means” in the first part of the claim that describes the known type of network upon which the improvement runs after being established when the “instructions for a given computer” recited in the preamble are “loaded into the given computer.”

<sup>17</sup> In their opening claim construction brief, Plaintiff describes the claimed subject matter as having the license server perform these functions: “In certain embodiments, the system can track software usage by allowing the computer acting as the license server to (1) identify the current nodes that are using the software product at a given time; (2) handle licensing terms for the

function (or even disclosing it) because there is no such support. Instead, Uniloc argues “computer memory” is inherent structure, but this argument fails because “computer memory” cannot perform the functions expressly linked to software running on a computer node acting as a license server. *See Microprocessor Enhancement Corp. v. Texas Instruments, Inc.*, 520 F.3d 1367, 1375 (Fed. Cir. 2011) (“[the claim] is clearly limited to a pipelined processor processing the recited structure and capable of performing the recited functions”).

Even if a license server must necessarily include memory, which can store data, memory does not “cause storage of the number of licenses available.” *See Northrop Grumman Corp. v. Intel Corp.*, 325 F.3d 1346, 1352 (Fed. Cir. 2003) (declining to identify as structure signals monitored by “means for monitoring”). A programmed computer performs that function. Computer memory is incapable of “identifying” or “determining.” These functions require logic and processing—capabilities of an actual program. The corresponding structure that performs the recited function is the license server running a license system “such as NetLS” and is not simply “computer memory.”

**B. Data Structure for a Policy Server Database, “Maintained Locally on the Given Computer”**

<b>Term/Clause</b>	<b>Plaintiff’s Construction</b>	<b>Defendants’ Construction</b>
“Maintained locally on the given computer”	No construction necessary, or  Stored at the given computer	Present on the user’s computer prior to the permission-to-run availability determination.

“Maintained locally” modifies the data structure for the database established at the client node that organizes and stores permission-to-run rules. *See, e.g.*, ‘222 Patent at 3:48-52 (describing the improvement as including “a policy server database maintained on each node, containing data

---

software product; and (3) determine whether the terms would still be satisfied if another node

specifying conditions under which usage of the software product is permitted on the corresponding node”). The parties dispute whether “maintained locally” connotes a temporal persistence (as Defendants contend) or simply restates the location of the database at the given computer (as Plaintiff contends). *See* Pltf. Br. at 5.

The intrinsic record consistently describes the policy server database structure as “maintained locally” over a period of time. Timing is a central aspect of the ’222 Patent’s *concurrent* licensing system and claim 18 in particular. *See, e.g.*, ’222 Patent at 3:34-42 (describing the invention as “identify[ing] the current set of nodes that are using the software product *at a given time* . . . to determine whether at any given time the conditions would still be satisfied” if another node is permitted to use the software product) (emphasis added).

Figure 5 (below) shows the construction of the policy server database structure:

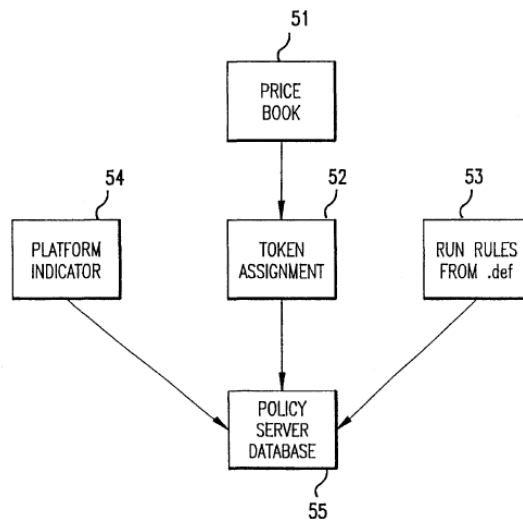


FIG.5

The accompanying description in the specification reinforces that the database persists for some time

---

was granted permission to run the software product.” Pltf. Br. at 1 (citing JA00012, at 3:34-46).

after its structure is built: “License prices are *initially* established by management decision in price book 51, which forms the basis for assigning token values (step 52) required for license grant.” ‘222 Patent at 8:19-28. Then, prior to a permission-to-run determination, “the policy server database file 14 is cycled through to determine the enabled base token . . .” ‘222 Patent at 9:26-29. “The Licensing System on the server is then called” and, ultimately, a license is granted or “a failure is returned.” *Id.* at 9:29-43. Depending on the Licensing System process, a license check or periodic permission-to-run determination is made periodically (e.g., “The implementation here generates a ping every 10 minutes”) and the daemon cycles through the policy server database and checks the License Server on the network to confirm continued permission to run. *Id.* at 9:55-65 (“i.e., the Licensing System reports that the license is still valid”). The policy server database persists through this process. *See also* Fig. 2 (disclosing “timer interval handler” for controlling periodic communication).

In claim 18, the database structure is “established” by “instructions when loaded into the given computer.” ‘222 Patent at 14:45-7. Consistent with the specification, the plain language of claim 18 requires the software daemon to make a permission-to-run availability determination “on the basis of applicable data from the license server and the policy server database.” ‘222 Patent at 14:60-1. Thus, the policy server database is a necessary predicate for making the permission-to-run availability determination. It must be present before the determination is made or claim 18 is inoperable.

If no database is present before the permission-to-run availability determination, as Plaintiff contends is possible, the recited subfunction (“interfacing with . . . the policy server database”) cannot be performed. In turn, no applicable data from the policy server database is obtained, and “making a permission-to-run availability determination . . . on the basis of applicable data . . . from

the policy server database” cannot be performed. Unless the database is established and maintained locally during the *get\_license* process, the functional requirements of claim 18 cannot be fulfilled.<sup>18</sup>

Plaintiff’s proposed construction “should be viewed with extreme skepticism” because it encompasses inoperable embodiments. *Talbert Fuel Sys. Patents Co. v. Unocal Corp.*, 275 F.3d 1371, 1376 (Fed. Cir. 2002).<sup>19</sup>

### C. “Policy Server Means”

Term/Clause	Plaintiff’s Construction	Defendants’ Construction
“Policy server means, maintained and operating locally, as an independent process, separate from the software product on the given computer, and in association with the policy server database, for (i) communicating with the license server, (ii) interfacing with both (aa) the software product and (bb) the policy server database, and (iii) making a permission-to-run availability determination, with respect to local usage of the software product, on the basis of applicable data from the license server and the policy server database”	<p><u>Function</u>: (i) communicating with the license server, (ii) interfacing with both (aa) the software product and (bb) the policy server database, and (iii) making a permission-to-run availability determination, with respect to local usage of the software product, on the basis of applicable data from the license server and the policy server database</p> <p><u>Structure</u>: a daemon computer program programmed according to</p>	<p><u>Function</u> : (i) communicating with the license server, (ii) interfacing with both (aa) the software product and (bb) the policy server database, and (iii) making a permission-to-run availability determination, with respect to local usage of the software product, on the basis of applicable data from the license server and the policy server database, so that enforcement of license terms applicable to the software product at the given computer is achieved on the basis of license policy maintained at the given computer as well as applicable data from the license server.</p> <p><u>Structure</u>: a daemon computer program (<i>i.e.</i>, a software program</p>

<sup>18</sup> In that case, “[i]f the policy server database file 14 cannot be successfully accessed in step 621 [of Fig. 6] to determine the relevant license rules (a matter checked in step 622),” the specification provides for an alternative process. ‘222 Patent at 9:13-15. If the system is in enforcement mode (as it is in claim 18), permission to run is denied based on the failure to access the database, putting this scenario outside the scope of claim 18. *Id.* at 9:15-17 and Fig. 6. Alternatively, if a reserved license is available, “access to the policy server database file 14 is skipped altogether,” putting this scenario outside the scope of claim 18 too. *Id.* at 9:17-20 and Fig. 6.

<sup>19</sup> Decision vacated and remanded on other grounds, 537 U.S. 802 (2002).

	the algorithm(s) described in Figures 1-7, 3:49-4:7, 7:17-9:45, and equivalents <sup>20</sup>	that runs in the background) programmed according to the algorithm described in Figures 1-4, 6 and 7 and associated description at 3:49-4:7, 7:17-8:10 and 8:31-9:45
--	---	--

The parties agree that the specification discloses a “daemon computer program” programmed to carry out a particular algorithm as the structure for executing the identified function of the “policy server means.”<sup>21</sup> Where a specification discloses a computer programmed to carry out an algorithm as the structure for executing a claimed function, the corresponding structure is limited to a computer programmed to perform the disclosed algorithm. *Harris Corp. v. Ericsson Inc.*, 417 F.3d 1241, 1253 (Fed. Cir. 2005) (a “computer-implemented means-plus-function term is limited to the corresponding structure disclosed in the specification and equivalents thereof, and the corresponding structure is the algorithm”); *WMS Gaming, Inc. v. Int’l Game Tech.*, 184 F.3d 1339, 1349 (Fed. Cir. 1999) (same). Accordingly, the parties agree that “policy server means” is limited to a “daemon computer program” programmed according to the algorithm(s) described in the specification.

The parties’ only disputes with respect to “policy server means” are: (i) whether the

<sup>20</sup> Plaintiff initially identified the structure as “a computer containing code for implementing a ‘policy server daemon’ as described in the patent, and equivalents” but subsequently changed their identification to “a daemon computer program programmed according to the algorithm(s) described in Figures 1-7, 3:49-4:7, 7:17-9:45, and equivalents.” See Pltf. Br. at 9.

<sup>21</sup> In addition to being limited by the algorithmic structure disclosed in the specification, the corresponding structure of policy server means (*i.e.*, “a daemon computer program...”) also includes additional structural limitations as reflected by the language of Claim 18. As described in the sections below, Claim 18 requires that the daemon computer program must be one that is also “maintained and operating locally, as an independent process, separate from the software product.” This means, *inter alia*, that the daemon computer program must be limited to one capable of servicing other software products.

technical term “daemon” requires explanation for the jury; and (ii) whether the policy server database is [part of the] corresponding structure for policy server means.

*1. “Daemon” is a Technical Term that Requires Explanation*

Because “daemon” is an unfamiliar technical term of art, explanation of the term is necessary to help the jury understand its meaning. *See, e.g., Abbott Labs. v. Sandoz, Inc.*, 544 F.3d 1341, 1360 (Fed. Cir. 2008) (claim construction “is for the purpose of explaining and defining terms in the claims, and usually requires use of words other than the words that are being defined” and that “claims are construed as an aid to the decision-maker, by restating the claims in non-technical terms”) (*citing Multiform Desiccants, Inc. v. Medzam, Ltd.*, 133 F.3d 1473, 1477 (Fed. Cir. 1998)). As such, Defendants’ propose that the term “daemon” means “a software program that runs in the background.” Plaintiff, on the other hand, proposes that the term “daemon” is a recognized term of art and requires no further construction.

The term “daemon” is not defined in the ‘222 Patent. In addition, the meaning of “daemon” is not well known to lay jurors. The jury would of course be mistaken in thinking, for example, that a daemon is a minor devil or evil spirit. There is thus potential to confuse the jury if no explanation is given to the term. Extrinsic evidence is permitted to assist the court in determining the meaning of a technical term.

The extrinsic evidence cited by Defendants’ confirms that their proposed definition reflects the meaning of the term “daemon.” A “daemon” is a software program that runs in the background. *See, e.g.,* Ex. 4 (American Telephone and Telegraph Co., *Unix Research System Programmer’s Manual*) at 52 (“a background process . . .”); Ex. 5 (Freedman, *The Computer Glossary*) at 115 (“Program that waits in the background ready to perform some action when a certain event occurs.”); Ex. 6 (Stevens, *UNIX Network Programming*) at 72 (“a daemon is a

process that executes ‘in the background [...] either waiting for some event to occur, or waiting to perform some specified task on a periodic basis’); Ex. 10 (Microsoft Press, *Computer Dictionary*) at 91 (“A program usually associated with UNIX systems that performs a utility function without being requested or even known of by the user. A daemon sits in the background and is called into play only when needed...”).<sup>22</sup>

Indeed, Plaintiff’s own extrinsic evidence also supports Defendants’ interpretation of “daemon” as “a software program that runs in the background.” *See* Pltf. Br., Ex. E at 91 (citing the same definition as Defendants’); *see also* Pltf. Br. Ex. F (“**daemon** A program that performs a single task and runs in the **background** ....” (emphasis added)). Plaintiff does not contest this interpretation proffered by Defendants,<sup>23</sup> and offers no evidence to the contrary that a daemon, by definition, operates as a background process.

2. *Defendants’ Proposed Construction Properly Identifies the Correct Corresponding Structure*

Plaintiff and Defendants’ agree that at a minimum, the proposed corresponding structure includes Figures 1-4, 6 and 7, 3:49-4:7, 7:17-8:10 and 8:31-9:4. Plaintiff proposes that the corresponding structure additionally includes Figure 5 (illustrating the structure of the policy server database) and the associated description in the specification at 8:20-30.

The claim element “policy server means” is composed of three identifiable functions which recite : (i) communicating with the license server, (ii) interfacing with both the software product and the policy server database, and (iii) making a permission-to-run availability

---

<sup>22</sup> All exhibits are attached to the Declaration of Timothy R. Shannon in Support of Defendants’ Joint Responsive Claim Construction Brief.

determination, with respect to local usage of the software product, on the basis of applicable data from the license server and the policy server database. ‘222 Patent at Claim 18(b). The agreed-upon structure, at a minimum, is “a daemon computer program.” The specification refers to the structure as the “local policy server daemon 16” or “daemon 16.” The “link” or between the local policy server daemon and the three functions is set forth in the ‘222 Patent at 3:53-65:

Each node also has a policy server "daemon" (which may be implemented in software) in association with the corresponding policy server database, for (i) communicating with the license server, (ii) interfacing with both the software product and the corresponding policy server database, (iii) making a permission-to-run availability determination, with respect to local usage of the software product, on the basis of applicable data from the license server and the corresponding policy server database, so that enforcement of license terms applicable to the software product at a given local node is achieved on the basis of both license policy maintained at such local node as well as applicable data from the license server.

Plaintiff’s proposal incorrectly identifies the policy server database as additional corresponding structure for the “policy server means.” This interpretation is incorrect. The specification does not disclose or clearly link the policy server database to the claimed functions as required by § 112 ¶6. Rather, the policy server database is recited as its own element in claim 18 and is merely referenced in the claim language pertaining to policy server means. The daemon, as is described above, is the only necessary and clearly linked structure for performing the claimed functions. *Dell USA L.P. v. Lucent Techs, Inc.*, 464 F.Supp.2d 620, 648 (E.D. Tex. 2006) (“corresponding structure must only be that which actually performs the specified function and not merely enables the corresponding structure to operate as intended”); *TV Interactive Data*

---

<sup>23</sup> Plaintiff stated “[U]niloc agrees that it would have been understood by one skilled in the art at the time of the patented invention that a daemon may comprise a ‘background process.’” Pltf. Br. at 11; Nettles Decl. at ¶ 17.

*Corp. v. Sony Corp.*, Civil Action No. 10-cv-0475, 2012 WL 6020113, at \*4 (N.D. Cal. Dec. 3, 2012) (“The corresponding structure, furthermore, includes only the features necessary to perform the claimed functions.”). A clear link or association between the disclosed structure and function recited in the claim is also required. *See Northrup Grumman Corp. v. Intel Corp.*, 325 F.3d 1346, 1352 (Fed. Cir. 2003) (“A court may not import into the claim features that are unnecessary to perform the claimed function.”).

Further, the policy server database is incapable of performing the claimed functions. *See Default Proof Credit Card System, Inc. v. Home Depot U.S.A., Inc. (d/b/a The Home Depot)*, 412 F.3d 1291, 1299 (Fed. Cir. 2005) (“structure disclosed in the specification must be clearly linked to and capable of performing the function claimed by the means-plus-function limitation”). The policy server database does not communicate with the license server, interface with the software product and undoubtedly does not interface with itself as required by the claim. The policy server database also cannot make a permission-to-run determination. A database, as a storage device, is incapable of performing these functions. To perform these functions, an element must have computing capabilities, like the daemon computer program described above.

The policy server database is acted upon by the claimed policy server means (the daemon); it is not the means itself. The daemon only interfaces *with* the database and makes a permission-to-run *on the basis of* the database. The database is the object of these functions, not the actual structure performing these functions. The correct identification of structure covers only those features that perform the recited function, regardless of whether there are additional elements directly tied to the claimed function. *See e.g., Northrop*, 325 F.3d at 1352 (determining “means for monitoring” does not include as structure the signal that it monitors). The database, acting as the object, is not performing the claimed functions and cannot be considered part of the

corresponding structure for the policy server means.

**D. “Maintained and Operating Locally”**

Term/Clause	Plaintiff’s Construction	Defendants’ Construction
“Maintained and operating locally”	Stored and operable at the given computer [as a process distinct from the software product]	Stored and operating on the given computer

As a preliminary matter, the parties disagree how the language of Claim 18(b) should be parsed. Plaintiff acknowledges that the three phrases, (1) maintained and operating locally, (2) as an independent process, and (3) separate from the software products, are “three . . . *independent* clauses.” Pltf. Br. at 14 (emphasis added). As a matter of simple grammar, the phrases should be read independently.

Plaintiff nevertheless invites the Court to mash the three limitations into one, on the theory that it would “simplify the task” of construction. *Id.* at 14-15. Plaintiff is inviting reversible error. Claim “simplif[ication]” is neither a legitimate basis nor a legitimate goal of claim construction. To eliminate detail is to eliminate limitations is to broaden scope. *Bicon, Inc. v. Straumann Co.*, 441 F.3d 945, 950 (Fed. Cir. 2006) (stating that “claims are interpreted with an eye toward giving effect to all terms in the claim”); *Elektro Instrument S.A. v. O.U.R. Scientific Intern., Inc.*, 214 F.3d 1302, 1307 (Fed. Cir. 2000) (adopting a construction of a claim term because “[a]ny other conclusion renders the [claim language] superfluous”).

In particular, Plaintiff’s proposed construction (*i.e.* “stored and operable at the given computer as a process distinct from the software product”) reads the limitation “separate from the software product” out of the claim. *See* Pltf. Br. at 13. This is impermissible. *Bicon*, 441 F.3d at 950.

It also confuses the basic rules of grammar. All three phrases do “modify the subject

‘policy server means’,” Pltf. Br. at 19, but that does not mean the three phrases all mean the same thing. Different adjectives – here, different verbal adjectives – denote different qualities. They have different meanings. By way of example: the words “little” and “black” do not mean the same thing just because they both modify “dress.” Each term describes a particular quality (size, color). And the term “little” may have a meaning peculiar to that art. The defendants are not inserting limitations, Pltf. Br. at 16-17; they are reading the plain text.

With respect to the first phrase, “maintained and operating locally,” both parties appear to agree that the phrase means that the daemon must be stored at the given computer. *See also* ‘222 Patent at Fig. 1 (showing the daemon on the “local system,”), 3:53 (“Each node also has a policy server ‘daemon’”), 3:39-40 (“The local policy server daemon operates at the computer node.”). Defendants are simply making clear that the daemon must also actually be *operating* on the node, as the plain text suggests.

#### **E. “As an Independent Process”**

<b>Term/Clause</b>	<b>Plaintiff’s Construction</b>	<b>Defendants’ Construction</b>
“as an independent process”	None given.	executing instructions as a process that is different from the software product’s process

To construe the phrase, “[operating] as an independent process,” we begin with the text. The phrase “as an independent process” is set off by commas; it is a subordinate clause directly linked to and modifying the verb, “operating.” The clause describes not *where* the daemon code is located (an issue addressed by the next clause) but rather *how* it operates. The clause makes clear that, when operating, the daemon must do so using a distinct process. It must

operate/run/execute independently.<sup>24</sup>

The specification reinforces the point. Figure 1 refers to a “secure path *between processes*.” ‘222 Patent at Fig. 1, Item 1(emphasis added). The software has one process and the daemon has another. The communication between the two, we are told, “is handled by an *interprocess* communication in Unix.” ‘222 Patent at 5:62-65 (emphasis added). Again, the software has one process and the daemon has another. The daemon is not only physically located outside the software, it operates using a distinct process.

The file history reinforces the point. The applicant distinguished Corbin on the ground that “in the case of the present invention, a single policy server means would *function* independently of all five applications.” JA00295 (emphasis added). The applicant’s disclaimer addressed not only location of the daemon (addressed below), but also the operation of that daemon. The claimed daemon would “*run* as a process independent of the other applications.” *Id.* (emphasis added).

Logic reinforces the point. If the daemon was part of the same process as the software product, then whenever the software product was not running (e.g. upon having been refused permission to operate), that combined-software-and-daemon process would shut down and the daemon could no longer serve as a middle man for the other software products. The preferred embodiment would not work.

Finally, extrinsic evidence reinforces the point. “A process is basically a program in execution.” Ex. 9 (Tanenbaum, *Modern Operating Systems*) at 12. That is, a process is an

---

<sup>24</sup> Notably, the rest of 18(b) uses verbs as well. The daemon is communicating, interfacing, and making a permission to run availability determination. These describe what the daemon is *doing*, not just *where* it sits physically with respect to other code.

instance of a program (a set of written instructions) being executed. Ex. 7 (Levine, *The UNIX Dictionary of Commands, Terms, and Acronyms*) at 198 (“program that is currently being executed”); Ex. 8 Silberschatz, *Operating System Concepts*) at 90 (“We emphasize that program by itself is not a process; a program is a passive entity . . . whereas a process is an active entity”); Ex. 11 Oxford University Press, *Dictionary of Computing*) at 358) (“A stream of activity”); Nettles Decl. at ¶ 22.<sup>25</sup> The word process refers to operation, to running.

In short, “[operating] as an independent process” means “executing instructions as a process that is different from the software product’s process.”<sup>26</sup>

#### **F. “Separate from the Software Product”**

<b>Term/Clause</b>	<b>Plaintiff’s Construction</b>	<b>Defendants’ Construction</b>
“separate from the software product”	None given.	not part of the software product and capable of servicing other software products

Finally, the Court should construe Claim 18 to require that the daemon sits outside any one software product and is instead available to all of them. The Court should do so for two reasons. First, as noted above, subsection (b) of Claim 18 is written as a means-plus-function claim limitation. The structure disclosed in the specification clearly shows a daemon that sits outside any one software product and is instead available to all of them. Second, subsection (b)

<sup>25</sup> Declaration of Scott M. Nettles in Support of Defendants’ Joint Responsive Claim Construction Brief.

<sup>26</sup> We note that Plaintiff has objected to the incorporation of “Process ID” into the definition of the disputed term. Defendants believe that process ID is a useful indicator for the jury, evidence of the actual, underlying independence. However, to the extent that a process ID is an indicator of independence rather than part of the definition, we are prepared to treat it is a fact question and remove it from our proposed construction.

also includes claim language (“separate from the software product”) that compels the same result – a daemon that services all of the software products on the node, not captive to any one of them.

### *1. 112(6) Analysis*

As noted above, the parties agree that the “policy server means” is a daemon, the structure of which must be set forth in the specification per § 112(6).

The specification of the ‘222 Patent expressly describes a communication module 21 inside the daemon program (*i.e.*, daemon 16) as handling “communication with the various software products.” ‘222 Patent at 7:17-22. That is, it handles communications with all of the *multiple* different software products on the node. *Id.* And Figure 1 could not be clearer; it shows the multiple software products serviced by the node’s local daemon. *See also id.* at 3:3-7 (describing the shortcomings of “ten different products” requiring “ten different substantial software portions” in the prior art); 5:33-34 (each of these [multiple] products] includes a call . . . to the local policy server daemon.”); 5:54-55 (daemon reporting to the “applicable [*i.e.* one of many] software product[s].”); *cf. also id.* at 3:66-4:2 (the database associated with the daemon contains rules for the “plurality of software products” on the node). In every instance, the daemon is described as being available to multiple software products, and able to service each of them. The structure – *i.e.* the thing – disclosed in the ‘222 Patent as performing the functions of Claim 18(b) is a daemon that is available to multiple software products.

### *2. Ordinary Analysis*

Even if the foregoing were not enough (it is), the applicant added a phrase to subsection (b) – tucked between “means” and the recitation of the functions associated with the means – that points to the same result.

We turn first to the text. The phrase “separate from the software product” connotes that

the daemon is not part of the software product. If the daemon code were part of the software products's code, it would not be separate. Looking further down in the text of Claim 18(b), the text states that the daemon must “interface[e] with both the (aa) the software product and (bb) the policy server database.” This, too, suggests that the daemon must be separate from the software product since it is interfacing with it.

The prosecution history removes all doubt. “[P]rosecution history disclaimer ‘limits the interpretation of claims so as to exclude any interpretation that may have been disclaimed or disavowed during prosecution in order to obtain claim allowance.’” *Sta-Rite Indus., LLC v. ITT Corp.*, 682 F.Supp.2d 738, 744 (E.D. Tex. 2010) (quoting *Omega Eng'g Inc. v. Raytek Corp.*, 334 F.3d 1314, 1323 (Fed. Cir. 2003)). “[P]rosecution history is another tool to supply the proper context for claim construction because a patent applicant may also define a term in prosecuting the patent.”) *Id.* (citing *Home Diagnostics, Inc. v. LifeScan, Inc.*, 381 F.3d 1352, 1356 (Fed. Cir.2004)).

Here, the applicants argued that their invention was different from (and superior to) Corbin because it permitted a single daemon to service multiple software products. JA00295. Unlike the software-product-specific modules in Corbin, this new daemon would exist on its own and be available to any and all of the (e.g. five) software products on the node. *Id.* The new daemon that sits outside any one software product and is instead available to all of them.

In sum, this limitation of “separate from the software product” – a limitation Plaintiff has quietly tried to read out of the patent with its “simplified” construction, Pltf. Br. at 19 – is both very real and very clearly defined in the patent: the daemon not part of the software product and is capable of servicing multiple software products.

## II. AGREED CONSTRUCTIONS

The parties jointly propose constructions for two terms, consistent with the ordinary meaning of the claim language, to provide helpful clarity for the jury. Subject to the Court's approval, the parties propose instructing the jury as set forth in the table below.

Term/Clause	Agreed Construction
“A digital storage medium encoded with instructions . . . the instructions causing administration of license terms”	A computer readable storage device containing instruction that, when executed, cause license terms to be enforced for a given computer
“Policy server database . . . containing data specifying conditions under which usage of the software product is permitted on the given computer”	A database [maintained on a given computer] containing information specifying the conditions under which the software product is permitted to be used on the given computer

## CONCLUSION

Defendants respectfully request that the Court adopt its proposed constructions, which are consistent with the intrinsic record and extrinsic evidence, will help the jury perform its infringement determination, and accurately reflect the statements and disavowals the inventors made to the patent office to obtain allowance.

Respectfully submitted,  
ALTAIR ENGINEERING, INC.  
By its attorneys,

/s/ Marshall MacFarlane  
Marshall MacFarlane, *pro hac vice*  
Christopher G. Darrow, *pro hac vice*  
YOUNG BASILE HANLON &  
MACFARLANE, P.C.  
3001 W. Big Beaver Rd., Ste. 624  
Troy, MI 48084-31070  
(248) 649-333  
[darrow@youmgbasile.com](mailto:darrow@youmgbasile.com)

Brian Craft, State Bar No. 04972020  
Findlay Craft, LLP  
6760 Old Jacksonville Hwy  
Suite 101  
Tyler, TX 75703  
(903) 534-1100  
[bcraft@findlaycraft.com](mailto:bcraft@findlaycraft.com)

Respectfully submitted,  
SLICKEDIT INC.  
By its attorneys,

/s/ Timothy Shannon  
Timothy R. Shannon, *pro hac vice*  
Seth S. Coburn, *pro hac vice*  
VERRILL DANA, LLP  
One Portland Square  
P.O. Box 586  
Portland, ME 04112-0586  
Tel: (207) 774-4000  
[tshannon@verrilldana.com](mailto:tshannon@verrilldana.com)

Collin M. Maloney, State Bar No. 00794219  
IRELAND, CARROLL & KELLEY, P.C.  
6101 S. Broadway, Suite 500  
Tyler, Texas 75703  
Tel: (903) 561-1600  
[cmaloney@icklawn.com](mailto:cmaloney@icklawn.com)

Respectfully submitted,  
ALTIUM, INC.  
By its attorneys,

/s/ Cabrach J. Connor  
David E. Dunham  
Cabrach J. Connor  
Miguel Rodriguez  
Jennifer Tatum Lee  
TAYLOR DUNHAM, LLP  
301 Congress Avenue, Suite 1050  
Austin, TX 78701  
(512) 473-2257  
[ddunham@taylordunham.com](mailto:ddunham@taylordunham.com)  
[mrodriguez@taylordunham.com](mailto:mrodriguez@taylordunham.com)  
[jtatum@taylordunham.com](mailto:jtatum@taylordunham.com)

Dated: November 12, 2013

**CERTIFICATE OF SERVICE**

I, Timothy R. Shannon, hereby certify that on November 12, 2013 I caused a true and accurate copy of this document to be served on counsel for the plaintiffs:

Edward E. Casto, Esq. (by ECF)  
NELSON BUMGARDNER CASTO, P.C.  
3131 West 7<sup>th</sup> Street, Suite 300  
Fort Worth, TX 76107  
Tel: (817) 377-9111  
[ecast@nbclaw.net](mailto:ecast@nbclaw.net)

James L. Etheridge (by ECF)  
ETHERIDGE LAW GROUP  
2600 E. Southlake Blvd.  
Suite 120-324  
Southlake, TX 76092  
Tel: (817) 470-7249  
[jim@etheridgelaw.com](mailto:jim@etheridgelaw.com)

T. John Ward (by ECF)  
J. Wesley Hill (by ECF)  
WARD & SMITH LAW FIRM  
P.O. Box 1231  
1127 Judson Road, Suite 220  
Longview, TX 75606  
Tel: (903) 757-6400  
[jw@wsfirm.com](mailto:jw@wsfirm.com)  
[wh@wsfirm.com](mailto:wh@wsfirm.com)

/s/ Timothy R. Shannon  
Timothy R. Shannon